

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-007407

(43)Date of publication of application : 12.01.1999

(51)Int.Cl.

G06F 12/00

G06F 13/00

(21)Application number : 10-117344

(71)Applicant : INTERNATL BUSINESS MACH CORP <IBM>

(22)Date of filing : 27.04.1998

(72)Inventor : BRENT ZION HAIRUPAAN
PETER KENNETH MALKIN
ROBERT JEFFREY SHUROSS
YU PHILIP S

(30)Priority

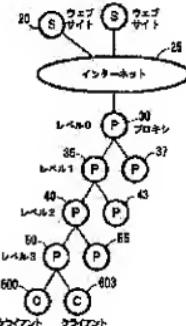
Priority number : 97 854226 Priority date : 09.05.1997 Priority country : US

(54) DYNAMIC PUSH FILTERING METHOD ACCCOMPANIED BY STAGING/ BUFFERING IN PROXY HIERARCHY

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a push base filtering system in a client-server hierarchy and a method for annexing the annotation of meta information related to the contents of a push object or urgency or both of them to the object based on using state information.

SOLUTION: Objects are staged in a server so that each filtered object can be quickly accessed at the time of requiring it thereafter. A contents provider or an upper level proxy for annotating an object transmits information including urgency, a summary or a title, group classification, or push discrimination, or all of them by using a PICS protocol so that using state or preference information related to the pushed object including using state information and user preference based on object group classification is transmitted to an upper level in hierarchy and each staged object is transmitted to a lower level in the hierarchy, improving the efficiency of caching.



LEGAL STATUS

[Date of request for examination] 23.10.1998

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

(51)Int.Cl.*

G 0 6 F 12/00
I3/00

識別記号

5 4 6
3 5 5

F I

G 0 6 F 12/00
13/005 4 6 K
3 5 5

審査請求 有 請求項の数27 O.L (全 18 頁)

(21)出願番号	特願平10-117344	(71)出願人	380009531 インターナショナル・ビジネス・マシーンズ・コーポレーション INTERNATIONAL BUSINESS MACHINES CORPORATION アメリカ合衆国10504、ニューヨーク州 アーモンク (番地なし)
(22)出願日	平成10年(1998)4月27日	(72)発明者	ブレント・ツイオン・ハイルバーン アメリカ合衆国10536-2010 ニューヨーク州カトナザ・テラス 28
(31)優先権主張番号	0 8 / 8 5 4 2 2 6	(74)代理人	弁理士 坂口 博 (外1名)
(32)優先日	1997年5月9日		
(33)優先権主張国	米国(USA)		

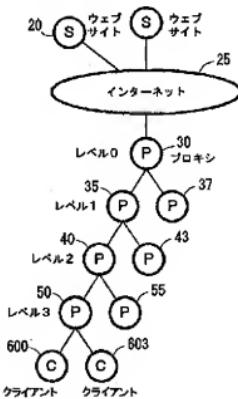
最終頁に続く

(54)【発明の名称】 プロキシ階層におけるステージング/バッファリングを伴う動的ブッシュ・フィルタリング方法

(57)【要約】

【課題】 利用状況情報に基づきクライアント-サーバ階層でのブッシュ・ベースのフィルタリング方式と、ブッシュ・オブジェクトにオブジェクトのコンテンツまたは緊急度あるいはその両方にに関するメタ情報の注釈を付ける方法を提供すること。

【解決手段】 オブジェクトをサーバ上でステージングし、フィルタリングされたオブジェクトが後に要求された時点で高速アクセスを可能にすることができる。PICSプロトコルを使って、コンテンツ・プロバイダまたはオブジェクトに注釈を付ける1位レベル・プロキシによって、緊急度、サマリまたはタイトル、グループ分類、もしくはブッシュの識別あるいはこれらのすべてを含む情報を送信し、オブジェクト・グループ分類に基づく利用状況情報およびユーザ・ブリファレンスを含む、ブッシュされたオブジェクトに関する利用状況またはブリファレンス情報を階層中を上方に送信し、それぞれのステージングされたオブジェクトを階層中を下方に送信してキャッシングの効率を向上させることができる。



プロキシ・サーバの階層

【特許請求の範囲】

【請求項1】 ブッシュされたオブジェクトが階層中を下方に送信されるプロキシ階層においてその後ブッシュされたオブジェクトを動的にフィルタリングする方法であつて、

ブッシュされたオブジェクトに関連する利用状況情報を階層中を上方に送信するステップと、

送信されたオブジェクト利用状況情報に基づいてその後ブッシュされたオブジェクトをフィルタリングするステップとを含む方法。

【請求項2】 ブッシュされたオブジェクトが階層中を下方に送信されるプロキシ階層においてブッシュされたオブジェクトを動的にフィルタリングする方法であつて、情報を総合しノード間で交換するステップと、ブッシュされたオブジェクトを、総合され交換された情報に基づいてフィルタリングするステップとを含む方法。

【請求項3】 前記フィルタリング・ステップがフィルタリングされたオブジェクトに関するメタ情報を送信するステップをさらに含む、請求項2に記載の方法。

【請求項4】 フィルタリング決定、選択された下位レベル・プロキシ/クライアントへのブッシュの成功裏の完了、利用状況情報、および階層内の他のプロキシ・ノードでのステージング決定のいずれかに基づいてオブジェクトの適応ステージングを行うステップをさらに含む、請求項1に記載の方法。

【請求項5】 ブッシュされたオブジェクトがメタ情報のコンテンツ階層を含み、その後ブッシュされたオブジェクトをフィルタリングする前記ステップがプロキシ階層中を下方にメタ情報を送信するステップをさらに含む、請求項1に記載の方法。

【請求項6】 メタ情報を送信する前記ステップが、オブジェクトの短い説明を階層中を下方に送信するステップと、フィルタリングされたオブジェクトをプロキシ階層内でステージングするステップとをさらに含む、請求項5に記載の方法。

【請求項7】 オブジェクトをステージングする前記ステップが、ユーザ・プロファイルで示された潜在的なインタレストを有するすぐ下位のレベルのすべてのプロキシまたはクライアント・ノードがオブジェクトを受信したか、あるいはステージング緊急度因数が所定および計算しきい値を下回ったか、そのいずれかの後にステージングされたオブジェクトをバージするステップをさらに含む、請求項4に記載の方法。

【請求項8】 前記フィルタリング・ステップがさらに階層内のすべての下位クライアント・ノードによるブッシュされたオブジェクトの総合利用状況およびプリファレンスを規定するステップをさらに含む、請求項1に記載

の方法。

【請求項9】 オブジェクトがオブジェクト・グループに分類され、各オブジェクト・グループの利用状況情報が以前のユーザ要求パターンに基づく、請求項4に記載の方法。

【請求項10】 利用状況情報が下位レベル・プロキシまたはクライアントの要求パターンを含み、

前記ブッシュされたオブジェクトを異なるグループに分類するステップと、

10 ブッシュされたオブジェクトのグループ分けを含む利用状況情報を総合しノード間で交換するステップと、総合され交換された情報およびブッシュされたオブジェクトのグループ化に基づいてブッシュされたオブジェクトをフィルタリングするステップとをさらに含む、請求項1に記載の方法。

【請求項11】 フィルタリング決定またはステージング決定が帯域幅、オブジェクト・プロパティ、またはクライアント特性のいずれかの閾値である、請求項4に記載の方法。

20 【請求項12】 クライアント特性がユーザ・プロファイルまたはプリファレンス情報を含む、請求項11に記載の方法。

【請求項13】 オブジェクト緊急度標識をブッシュされたオブジェクトに関連付けて階層中を下方に送信するステップをさらに含み、

前記フィルタリング・ステップがオブジェクト緊急度標識の閾値である、請求項1に記載の方法。

【請求項14】 緊急度標識を1つまたは複数の異なる階層レベルに関連付けるステップをさらに含み、

30 前記ステージング・ステップが、緊急度標識の閾値としてコンテンツ階層の1つまたは複数のレベルでオブジェクトをステージングするステップを含む、請求項4に記載の方法。

【請求項15】 前記フィルタリング・ステップとステージング・ステップのいずれかがオブジェクト・サイズの閾値である、請求項4に記載の方法。

【請求項16】 前記フィルタリング・ステップと前記ステージング・ステップのいずれかがオブジェクトの寿命または満了時間の閾値である、請求項4に記載の方法。

【請求項17】 前記ステージング・ステップに応答して、ブッシュされたオブジェクトのステージング・ステータスをブッシュされたオブジェクトへ送信するステップをさらに含む、請求項4に記載の方法。

【請求項18】 プロキシ階層が、前記フィルタリング・ステップと前記ステージング・ステップのいずれかが階層内のすべてのサーバによっては実行されない異種プロキシ構造を含む、請求項4に記載の方法。

【請求項19】 PICS利用状況ラベルを作成し、PICSカタゴリ値を使ってプロキシ階層の下位レベルでの総合利用状況情報を表すステップをさらに含み、

前記送信ステップがPICS利用状況ラベルを使って総合利用状況情報を階層中を上方に送信するステップを含む、請求項2に記載の方法。

【請求項20】PICSステージング・ラベルを作成し、PICSカテゴリ値を使ってプロキシ階層の所与のレベルにおけるステージングされたオブジェクトのステージング・ステータスを表すステップと、PICSステージング・ラベルを使ってステージング・ステータスを階層中を下方に送信するステップをさらに含む、請求項4に記載の方法。

【請求項21】PICSブッシュ・ラベルを作成し、PICSカテゴリ値を使ってブッシュされたオブジェクトの緊急度標識を表すステップと、PICSブッシュ・ラベルを使って緊急度標識を階層中を下方に送信するステップとをさらに含む、請求項4に記載の方法。

【請求項22】メタ情報プロトコルを使って階層中を情報を送信する、請求項1ないし3、5、6、9ないし10、14ないし15、19ないし20または21のいずれか一項に記載の方法。

【請求項23】PISCプロトコルを使って階層中を情報を送信する、請求項1ないし3、5、6、9ないし10、14ないし15、19ないし20または21のいずれか一項に記載の方法。

【請求項24】PICSブッシュ・ラベルを作成し、PICSカテゴリ値を使ってブッシュされたオブジェクトのサマリを表すステップと、PICSブッシュ・ラベルを使ってサマリを階層中を下方に送信するステップとをさらに含む、請求項4に記載の方法。

【請求項25】階層外のプロキシ・ノードについてのステージング決定に基づいてオブジェクトの適応ステージングを行うステップをさらに含む、請求項1に記載の方法。

【請求項26】ブッシュされたオブジェクトのストリームが階層中を下方に送信されるプロキシ階層において、その後ブッシュされるオブジェクトを動的にフィルタリングする方法であって、

ブッシュ・ストリームにメタ情報の注釈を付けるステップと、前記の注釈付けてステップに応答して、1つまたは複数のブッシュされたオブジェクトをフィルタリングするステップとを含む方法。

【請求項27】フィルタリング決定とステージング決定のいずれかが、コンテンツ階層のレベルが異なると変わる、請求項6に記載の方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、改良されたデータ処理システムに関する。本発明の特定の態様は、プロキシ・サーバの階層内でオブジェクトまたはドキュメントを送達する動的ブッシュ（または回送通信）フィルタリング方式に関する。本発明のさらに特定の態様は、ウェ

ブ・オブジェクトをワールド・ワイド・ウェブ（WWW）にブッシュする手順に関する。

【0002】用語集

本明細書で使用する用語の一部は辞書にある意味もあるが、次の用語集も役立つであろう。

【0003】インターネット

一連のTCP/IPプロトコルを使用するネットワークおよびゲートウェイからなるネットワーク。

【0004】クライアント

10 クライアントとは、コマンドをサーバに発行してそのコマンドに対応するタスクを実行させるコンピュータである。

【0005】サーバ

他のコンピュータのコマンドに従ってタスクを実行する任意のコンピュータがサーバである。ウェブ・サーバは通常1つまたは複数のクライアントをサポートする。

【0006】ワールド・ワイド・ウェブ（WWWまたはウェブ）

強調表示された言葉や関心がある語句（ハイバーリンク）をクリックすることにより、サーバ間およびデータベース間でインターネット・スイッチ上で情報を探すためのインターネットのアプリケーション。インターネットWWWサーバはクライアントをサポートし、情報を提供する。ウェブはすべての資源がURLとしてアドレス指定され、HTMLを使ってURLに対応する情報を表示し、他のURLへのポイント・アンド・クリックによるインターフェースを提供するインターネットと見なすことができる。

【0007】ユニバーサル・リソース・ロケータ（URL）

インターネット上の情報を一意的に識別またはアドレス指定するための方法。Eメール・アドレスのウェブ・ドキュメント・バージョンまたは完全に資格があるファイル名。これらはハーバーリンクでアクセス可能である。URLの1例は、「http://www.philippe.com:80/table.html」である。ここでURLは4つの成分を備える。左から始めて、最初は使用するプロトコルを指定するもので、残りのロケータと“：“で分離される。2番目は目標ホストのホスト名またはIPアドレスである。これは左側が「/」で、右側が「/」または任意選択として：“”で区切られる。ポート番号は任意選択で、左側はホスト名と「：」で区切られ、右側は「/」で区切られる。第4の成分は実際のファイル名またはプログラム名である。この例では、「.html」の拡張子がHTMLファイルであることを示している。

【0008】ハイパーテキスト・マークアップ言語（HTML）

HTMLはとりわけ、ドキュメントを作成し、ウェブ・クライアントから閲覧できる他のウェブ・ドキュメント

に（ハイバーリング経由で）接続するためにウェブ・サーバが使用する言語である。

【0009】ハイパーテキスト転送プロトコル（HTTPまたはhttp）

HTTPはクライアントからサーバへのすべての要求が独立して扱われることを意味する無状態プロトコルの例である。サーバは以前の接続の記録を持たない。URLの冒頭にある「http:」は要求元クライアントと目標サーバが指定の資源に関してHTTPを使って通信を行なうことを示している。

【0010】インターネット・ブラウザまたはウェブ・ブラウザ
httpなどのインターネット・プロトコルを実行し、その結果をユーザーの画面に表示するクライアント・インターフェース。ブラウザは、ユーザーがインターネットの「サーフィン」をする際に使用されるデスクトップ画面、ディレクトリおよび検索ツールを備えたインターネットのツール・ガイドとして機能できる。この適用例ではウェブ・ブラウザはワールド・ワイド・ウェブと通信するクライアント・サービスである。

【0011】クライアント・キャッシュ

クライアント・キャッシュは通常、クライアントがアクセスするオブジェクト用の1次キャッシュとして使用される。WWW環境では、クライアント・キャッシュは通常ウェブ・ブラウザによって実施され、現在の呼び出し中にオブジェクトをキャッシュする非持続性キャッシュでもよく、複数の呼び出しにまたがってオブジェクトをキャッシュすることもできる。

【0012】キャッシング・プロキシ

クライアントのためにエージェントとして働いて、オブジェクトのキャッシュされたコピーを見つける、ネットワーク内の専用サーバ。キャッシング・プロキシは、クライアント・キャッシュからのキャッシュ・ミスの結果として呼び出されたため、通常は2次またはそれ以上のレベルのキャッシュとして動作する。

【0013】HTTPデーモン（HTTPD）

ハイパーテキスト転送プロトコルおよび共通ゲートウェイ・インターフェース機能を備えるサーバ。HTTPDは通常、インターネット上のマシンへのハードウェア接続、およびTCP/IPカーリングなどインターネットへのアクセスを提供する、アクセス・エージェントによってサポートされる。

【0014】ワールド・ワイド・ウェブ（WWWまたはウェブ）の人の激しい高まりに伴って、インターネット上のトラフィック量も増加している。その結果、ウェブはネットワーク・パフォーマンスの主要なボトルネックになってしまった。低速ネットワーク・リンクを介してサーバに接続されているユーザーからドキュメントまたは情報の要求があった場合、ユーザー・エンドではかなりの待ち時間を覚悟しなくてはならない。要求されたドキュメ

ントを「ブル」するのに長時間待つという手間を避けるための代替方法は、コンテンツ・プロバイダに該当するドキュメントが使用可能になると同時にあらかじめ規定したユーザー・プリファレンスまたはプロファイルに基づいてユーザーへドキュメントを「プッシュ」させる方法である。

【0015】このプッシュ方法は、ネットワークからオーバーフローする傾向がある。これは、ユーザーのプリファレンス指定が不適切になりがちなため、あまりにも多くのドキュメントがユーザーの元にプッシュされることが原因で起こる。

【0016】従来の「ブル」方法では、アクセスの待ち時間を減らす1つの方法は人気があるドキュメントまたはユーザーに身近な情報のコピーをキャッシュすることで、そこからのアクセスの待ち時間はより短くなる。このキャッシングはネットワーク上でのさまざまなポイントで実施できる。例えば、大きな大学や会社では独自のローカル・キャッシングがあり、そこからネットワークの加入了すべてのユーザーがドキュメントを取り出すことができる場合がある。場合によっては、クライアントのためのエージェントとして動作できる、キャッシング・プロキシと呼ばれる専用サーバが、ドキュメントのキャッシュされたコピーを見つけるためにネットワーク中で実施される。通常、キャッシング・プロキシは（1次）クライアント・キャッシングからのキャッシング・ミスのみ関連しているため、2次またはそれ以上のレベルのキャッシングとして動作する。クライアント・キャッシングは通常ウェブ・ブラウザの一部であり、現在の呼び出し中にアクセスされたオブジェクトを記憶する（Mosaićで実施されるような非持続性キャッシング）、または、複数の呼び出しにまたがってアクセスされたドキュメントを記憶することができる。

【0017】一般的に、プロキシの階層はクライアントとサーバ（1つまたは複数の）によって構成される。コンピュータ・ネットワークでは、プロジェクト・プロキシ、部門プロキシ、サイト・プロキシなどが1つまたは複数存在する。インターネットのサービス・プロバイダは各近隣、各下位地域、各地域などの1つまたは複数でプロキシを実施できる。クライアントまたはプロキシあるいはその両方がキャッシング階層を形成する。厳密な階層では、キャッシング・ミスが生じると（クライアント）またはプロキシは、CERN HTTPキャッシングで使われるHTTPインターフェースのようなキャッシング・プロキシ・インターフェースを介して階層のすぐ上位レベルから欠落したオブジェクトを要求する。より最近では、ハーベストにおいて、キャッシング・ミス時に「兄弟」または「近隣」キャッシングに問い合わせせることができる（C. M. ブラウン（Brown）他、「Harvest: A Scalable, Customizable Discovery and Access System」）。

5 ロード・ドライブ、コンピュータ科学部、技術レポート

C U - C S - 7 3 2 - 9 4 、 1 9 9 4 年を参照)。いずれの場合も、キャッシングの決定は他のプロキシでキャッシングされたオブジェクトとは無関係にそれぞれのローカル・プロキシで行われる。言い換えると、キャッシングの決定はローカル・キャッシュのコンテンツまたはオブジェクト特性あるいはその両方の間数としてのみ実行される。

【0018】

【発明が解決しようとする課題】以上のように、プッシュ・ベースのフィルタリングおよびプロキシ・サーバを開発し、ビューウの実際の利用状況に基づくシステムが必要である。さらに、ステージング決定がプッシュ・フィルタリング決定およびプッシュ・アクティビティの結果に基づいて実行されるシステムおよび方法が必要である。また、プロキシ・サーバ間、コンテンツ・サーバとクライアント間での情報の送受信または交換によってプロキシ階層により効果的に適用する必要がある。本発明は上記の必要に関する。

【0019】

【課題を解決するための手段】上記の必要に応じて、本発明は、実際の利用情報を基づくクライアント-サーバ階層でプッシュ情報をフィルタリングする方法およびシステムを対象とする。利用状況情報は実際のオブジェクト参照/アクセス・パターンを含むことができる。ステージング・キャッシング・マネージャはサーバ(1つまたは複数)で実施され、フィルタリングされたオブジェクトが後で要求された時点で高速アクセスを可能にできる。

【0020】プッシュされたオブジェクトが階層を下方に送信されるプロキシ階層においてプッシュされたオブジェクトを動的にフィルタリングする本発明による機能を備えた方法は、プッシュされたオブジェクトに関連する利用情報を階層を上方に送信するステップと、その後、送信されたオブジェクト利用情報に基づいてプッシュされたオブジェクトをフィルタリングするステップを含む。

【0021】本発明の別の態様は、情報を総合してノード間で交換する追加ステップと、プッシュされたオブジェクトを、総合され交換された情報に基づいてフィルタリングする追加ステップを含む。このフィルタリング・ステップは、さらにフィルタリングされたオブジェクトに依存するメタ情報を送信するステップを含むことができる。

【0022】本発明のさらに別の態様は、フィルタリング決定、選択された下位レベル・プロキシ/クライアントへのプッシュの成功裏の完了、利用状況情報、および階層内の他のプロキシ・ノードでのステージング決定のいずれかに基づいてオブジェクトの適応ステージングを行うステップを含む。その他の機能によってプロキシ・サーバはステージングされたオブジェクトをタイムリにバージし、オブジェクトの冗長ステージングを削減する

ことによってより効率的にキャッシングを管理することができる。

【0023】本発明のさらに別の態様では、プッシュされたオブジェクトがメタ情報のコンテンツ階層を含み、その後プッシュされたオブジェクトをフィルタリングするステップがプロキシ階層中を下方にメタ情報を送信するステップをさらに含む。

【0024】本発明の別の態様では、プッシュ・フィルタリングはコンテンツ・レベルでのプッシュの決定を含む。フィルタリングの決定はタイトルまたはサマリ(コンテンツ全体ではなく)だけを次の(下位の)レベルの階層のいくつかのノードへプッシュすることでもよい。このように、フィルタリングの決定は次のどの(下位の)レベルのノードがプッシュを受信するかだけでなく、各ノードが受信するコンテンツ・レベルで可能である。次の(下位の)レベルのノードへのフィルタリングの決定はその次の(下位の)レベルのノードの下のすべてのユーザの利用状況の総合情報を基づくことができる。

【0025】本発明のさらに別の態様では、プッシュされたオブジェクトのストリームが階層を下方に送信され、その後プッシュされたオブジェクトを動的にフィルタリングする方法が、プッシュ・ストリームにメタ情報の注釈を付けるステップと、注釈付けのステップに応答してプッシュされた1つまたは複数のオブジェクトをフィルタリングするステップを含む。

【0026】本発明のさらに別の態様では、インターネット環境で、P I C Sプロトコルを使って各種の情報を送信できる。まず、コンテンツ・プロバイダまたは上位レベルのプロキシ・ノードはP I C Sを使ってオブジェクトに社旗を付ける、すなわち、プッシュ・オブジェクトに関する情報の特性を指定できる。この情報はプッシュ・オブジェクトの緊急度または優先度、オブジェクト・コンテンツのサマリまたはタイトル、グループ分類、またはプッシュの送信元チャネル(識別)あるいはこれらすべてを含むことができる。第2に、P I C Sを使ってプッシュ・オブジェクトのプリファレンス情報の利用状況を階層の下位レベルから上方に送信できる。これにはオブジェクト・グループ分類に基づく利用状況情報およびユーザ・プリファレンスを含むことができる。第3に、P I C Sを使って、それぞれのステージングされたオブジェクト(上位階層のいくつかまたはすべての)のステージング状態を階層を下方へ送信することができる。この場合、P I C Sプロトコルを一般化して階層全体にわたって情報の交換または送信を行うことができる。より詳細に言えば、これらの各種の情報はP I C Sプロトコルを使ってオブジェクトのヘッダに記憶することができる。各情報タイプについて新しいP I C Sレベルを定義してP I C Sカテゴリ値を特定の状況に応応させることができる。下位レベル・サーバ(またはクライ

アント) は P I C S カテゴリ値を解釈してブッシュまたはステージングの決定を効率化することができる。

【0 0 2 7】

【発明の実施の形態】図1に本発明の機能を備えたプロキシ・サーバの階層の全体アーキテクチャを示す。図に示すように、クライアント(6 0 0 . . . 6 0 3)はプロキシ・サーバ(3 0 . . . 5 5)の階層(レベル0 . . . レベル3)を介してインターネット(2 5)に接続できる。この例に限ってはとくに、図示のプロキシ階層は4つのレベルのプロキシ・サーバを含む。プロキシ階層のレベルはいくつでもかまわず、クライアント(6 0 0 . . . 6 0 3)は実際そのどのレベルにも接続できることを当業者なら理解するであろう。最高レベル(レベル0)のプロキシ(3 0)はインターネットに接続する。レベル1のプロキシは2つ(3 5と3 7)あり、一方のレベル1プロキシ3 5はレベル2のプロキシ(4 0と4 3)に接続される。クライアント6 0 3はそのすぐ上のレベル(レベル3)のプロキシ(5 0)に、次いでレベル2、1、0のプロキシ(4 0、3 5、3 0)に結合される。クライアントは、インターネット(2 5)を介して様々なウェブ・サイト(2 0)にアクセスすることができる。クライアントはステージングおよびユーザ情報を管理するためそれ自体のクライアント・プロキシを有することができる。プロキシ・サーバの階層は1つのクライアント(6 0 0)と単一のプロキシ・サーバを含むこともできることを当業者なら理解するであろう。

【0 0 2 8】クライアント(6 0 3)から見ると、ある種のプロキシ(5 5、4 3、および3 7)はそのプロキシ階層の一部ではない。代表的な通信パスがすぐ上位レベルのプロキシ群であるが下位レベルのプロキシは直接インターネットの他の上位レベルのプロキシへ通信が可能であることを当業者なら理解するであろう。

【0 0 2 9】従来のプロキシ階層では、ブッシュされたオブジェクトの受信後、オブジェクトは次の(下位の)レベルのプロキシへ下にブッシュされる。他方、ローカルに使用できないオブジェクトでの「ブル」要求の場合、欠落オブジェクトの要求は次の上位レベルのブッシュへ対して行われる。上位レベル・プロキシが以前にオブジェクトをステージングした場合、このプロキシはオブジェクトを下方に渡す。そうでない場合、このプロキシは上位レベル・プロキシからオブジェクト入手しようとすると。入力されたオブジェクトはそのオブジェクトを要求している次の下位レベルのプロキシへ下方に渡される。

【0 0 3 0】図1はオブジェクトおよび要求の情報のフローを表す論理接続だけを示す図であることに注意されたい。この図は物理的接続図ではない。論理接続は作業負荷やノードまたはリンク障害などの物理的イベントによって変わる場合がある。オブジェクトのタイプが異なれば、ユーザへ到達する論理パスも異なる場合がある。

【0 0 3 1】概観すると、本発明のサーバ(3 0 . . . 5 0)は、オブジェクトと共にブッシュ・オブジェクトおよびそのステージングの決定に関する情報を次の(下位の)レベルのプロキシへブッシュする。実際、それ自体のステージングの決定だけでなく、上位階層全体のステージングの決定(1つまたは複数の)を下方に渡すことができる。

【0 0 3 2】本発明は階層の上下間に情報を効率的に送信するための機能をさらに備える。h t t p を実施する場合、情報交換は既存のウェブ・プロトコルを使ってオブジェクト・ヘッダ内に含めることができる。P I C S(「インターネット・コンテンツ選択用プラットフォーム」)は電子コンテンツに関するメタ情報を送信する方法を規定する。P I C S はウェブ・コンソーシアム・プロトコル勧告である(h t t p : // www. w3. org / P I C S を参照)。P I C S は「このコンテンツにどれくらいの量の n u d i t y (ステディティ) が関連付けられているか」などの値ベースの評価ラベルを界隈するために初めて使用されたが、メタ情報のフォーマットと意味は完全に一般的である。P I C S では、電子コンテンツのメタ情報が「評価サービス」または情報の生成源および予定利用状況に従ってグループ分けされそのグループの1つの中で情報の任意の数のカテゴリまたはディメンションを送信できる。各カテゴリには許される値の範囲があり、特定の1件のコンテンツについて、特定のカテゴリが単一または複数の値をとることができる。さらに、メタ情報グループ(「P I C S ラベル」として知られている)は満了情報を含むことができる。また、P I C S は複数の電子コンテンツに適用する機能である。特定の1件の電子コンテンツのP I C S ラベルは独立してコンテンツに追加またはそこから削除できる。

【0 0 3 3】例えば、「評価サービス」フィールドが「S a f e s u r e」評価システムに従って値ベースの評価ラベルを含むことを示す單一のP I C S ラベルを受けたサーバへイメージ・ファイルを送信することができる。本発明によれば、イメージ・ファイルは企業プロキシ通過する際に「評価サービス」フィールドがプロキシ・ステージング情報を含むことを示す第2のP I C S ラベルを受信することができる。専門プロキシを通過する際に第2のP I C S ラベルをはがすことができる。このように、クライアント・コンピュータは第1のP I C S ラベルだけを参照することができる。h t t p プロトコルはP I C S をサポートする要求ヘッダおよび応答ヘッダを増強している。N N T Pなどの他の共通アプロケーション・プロトコルを規定する技術機関もP I C S サポートを追加することを考慮している。こうしたプロトコルの一部として、要求によって、所望のP I C S ラベルのタイプのリストを含めることもできる。またP I C S は中央のラベル局サーバからP I C S 情報を受信する際会フォーマットを規定している。P I C S ラベルの例

10
10
30
30
40
40
50

11

は、(PICS-1. 1 "http://the.rating.service" label for "http://the.content" exp"1997.07.01T08:15-0500"r (n 4 s 3 v 10)) で、ここで「n」、「s」、「v」、「r」は各種メタ情報タイプの送信名、このコンテンツの適用可能な値は4 (nの場合)、3 (sの場合)、2 (vの場合)、および0 (rの場合)である。ID「http://the.rating.service」を認識するソフトウェアだけがこれらのカテゴリおよび値の解釈が可能であろう。

【0034】好みの実態形態では、3つの異なるPICSラベルが使用される。ブッシュ・ラベルまたは(P-ラベル)と呼ばれる第1の種類のPICSラベルはコンテンツ・プロバイダまたは上位レベルのプロキシが使用してオブジェクトの注釈付け、すなわち、ブッシュ・オブジェクトの特性を指定したりそれにに関する情報を指定する。このラベルは次のカテゴリのいずれかの組み合わせでも済むことができるがそれだけには限界はない。

【0035】・緊急度カテゴリ：「緊急度」カテゴリの値はオブジェクトを下方にプッシュする緊急度を示す。そのカテゴリ値として「UV」を定義する。

【0036】・サマリ・カテゴリ：「サマリ」カテゴリの値はブッシュ・オブジェクト/ドキュメントの短いサマリを示す。より一般的なケースでは、1つのオブジェクトを複数の詳細レベルで指定できる。このコンテンツ階層は上記の3つ以上のレベル（フル・コンテンツおよびサマリ）からなることができる。例えば、別のタイトル・レベルを含むことができる。ニュース・オブジェクトへの「テロリスト」がショッピング・センタに爆弾をしかけて2名が死んだ。」のサマリとニュースの全コンテンツである「爆弾の爆発」などのタイトルを含むことができる。それぞれの追加のコンテンツ・レベルの別のカテゴリを使ってタイトルなどを追加のコンテンツ・レベルをドーラベルで指定できる。この例には、オブジェクト・タイトルの「タイトル」カテゴリがある。さらに、サマリ・カテゴリに全オブジェクト（すなわち、全コンテンツ）からの異なる緊急カテゴリ値を指定もできる。例えば、より高い緊急度カテゴリ値をサマリに指定できる。

【0037】・グループ・カテゴリ：「グループ」カテゴリの値はプロジェクトの分類を示す。例えば社内の同報通信の場合、代表的なグループ・カテゴリは「企業情報」、「HR」、「部門」、「競合他社」などを含むことができる。グループ・カテゴリを導入する第1の目的は、ニーザ情報（次節で説明する）をグループ・カテゴリが収集できるようにし、グループ・カテゴリによるプッシュを可能にすることである。

【0038】・データ・カタログ：「データ・カタログ」

【0038】・チャネル・カナゴリ：「チャネル」カナゴリの値は同報通信チャネルまたはコンテンツ送信源を

示す。例えば、内部企業チャネル、またはPOINTCAST(<http://www.pointcast.com>)の商標でPointcast, Inc. から利用できる外部チャネルである。異なるチャネルは異なるグループ、カテゴリを備えることができる。

【0039】好ましい実施形態では、1つのグループ・カテゴリおよび1つのチャネル・カテゴリが形態を簡素化すると議論される。当論者は複数のグループまたはチャネル・カテゴリあるいはその両方への一般化は容易で10ある一チャネルごとに別々にグループ単位で統計情報が保持され、ブッシュ・ファイルタリングの決定がオブジェクト・グループに対して利用状況に基づいてなされるためることを理解するであろう。

【0040】ユーザー・ラベル(U-ラベル)と呼ばれる第2の種類のPICSラベルを使って階層の下位のレベルから上方にブッシュ・オブジェクトの利用またはブリファレンス情報を伝送できる。このラベルは次のカテゴリのいずれかの組み合わせも含むことができるがそれだけには限定されない。

20 【0041】・利用状況(usage)カテゴリ:「利用状況」カテゴリの値はオブジェクト(オブジェクト・グループ内の)はどのくらいの頻度で下位階層で参照/アクセスされているかを示す。このカテゴリ値は「RV」で示される。

【0042】・プリファレンス・カテゴリ：「プリファレンス」カテゴリの値はユーザがプロファイル指定によって指定する興味がある対象を示す。プロファイル情報はユーザの興味が変わって変更しなくなると廃止される。このカテゴリ値は「PV」で示される。

30 【0043】「ステージング」ラベル（C-ラベル）と呼ばれる第3のタイプのPICSラベルはプロキシがオブジェクトが階層を通過する際に動作情報（キャッシング／ステージング情報などの）を送信し、共用するためを使用する。このラベルは次のカテゴリのいずれの組み合わせも含むことができるがそれだけには限定されない。

【0044】・ステータス・カテゴリ：「ステータス」カテゴリの値はオブジェクトが上位カテゴリ内でステージングされるかまたはステージングの方法あるいはその両方を示す。このラベルはそれぞれの上位レベル隣層でオブジェクトがステージングされるかどうかを指定できる。コンテンツ階層が使用できる場合、カテゴリ値は全ドキュメントとサマリのどちらをステージングするかをさらに示すことができる。

【0045】オブジェクトのステーリング・ステータスは「CV」で示される。任意のプロキシでCV値を決定する方法の1つを以下に説明する。例えば、2進表記値を使って次のようにCV値を決定できる。n番目のレベルのプロキシの場合、下方に記されるオブジェクトのCV値はn個のビットを備え、k番目のビットには $(n -$

k) レベルのプロキシが階層内を下方にオブジェクトを送信した際にオブジェクトをステージングした場合は1となる。そうでない場合、k番目のビットは0になる。さらに、ステージング・ステータス情報を使って、オブジェクトをすぐ上位のレベルのプロキシに要求せずに、オブジェクト要求をオブジェクトをステージングした可能性が高い最も近い上位のレベルのプロキシへ転送することもできる。

【0046】図1を再度参照して、例えばオブジェクトAのCV値を考える。レベル0プロキシ(30)とレベル2プロキシ(40)はオブジェクトAをステージングし、レベル1プロキシ(35)はオブジェクトAをステージングしていないものとする。レベル3プロキシ50へ下方に渡されたオブジェクトAのCV値はこの場合「101」(2進値)または5(10進値)になる。プロキシ35オブジェクトプロキシ40へ下方に渡されたオブジェクトAのCV値はそれぞれ「1」と「10」である。階層内でステージング・ステータスを表す別の方法もあることを当業者は理解するであろう。これより簡素ではあるが正確さが低い方法として、單一のビットを使っていざれかの上位レベルのプロキシがオブジェクトをステージングしたかどうかを表す方法がある。ビットがオンの場合、上位レベル・プロキシがオブジェクトをステージングしている。そうでない場合、どの上位レベルのプロキシもオブジェクトをステージングしていない。

【0047】図2に図1のプロキシ・サーバのアーキテクチャの詳細例を示す。概例として、プロキシ・サーバはCPU(200)、ディスク(205)、持続性データまたはプログラム/コード混在論の磁気、電子、または光記憶メディア、およびCPU(200)によるデータまたはプログラムあるいはその両方の動的アクセスまたは実行あるいはその両方を行なうためのメモリ(207)を含む。本発明の精神と範囲を逸脱することなく、メモリ(207)内にインスタンス化された1つまたは複数のコンポーネントがディスク(205)、ネットワーク(25)他のサーバから直接アクセスして保守でき、複数のサーバに配布できることを当業者は理解するであろう。好ましくはCPU(200)上で実行可能なソフトウェアとして具体化されたプロキシ・サーバの3つの主要コンポーネントは、ブッシュ・ハンドラ(220)、欠落ブッシュ・オブジェクト要求ハンドラ(240)、およびヘッド要求ハンドラ(260)である。これらのコンポーネントについてはそれぞれ図5、10および12を参照しながら以下に詳述する。

【0048】メモリ(207)は本発明の機能に開示したいいくつかのその他の構造を含む。図8を参照しながら以下に詳述するように、キャッシング(270)は各プロキシ・ノードで保守される。ブッシュがフィルタリングされた場合にキャッシングを使ってブッシュ・オブジェクトをステージングする。オブジェクトを後に要求する場

合、この手順でアクセス時間が短縮される。キャッシングがディスク(205)などの記憶階層の下位レベルまで拡張できることに注意されたい。したがって、プロキシ内のキャッシングまたはステージングされたオブジェクトは階層のどのレベルにも常駐できる。図4を参照しながら以下に詳述するように、それぞれの次のレベルのノードの総合ユーザ情報はユーザ情報280が示すように保守されてフィルタリングの決定を支援する。図8を参照しながら以下に詳述するように、それぞれのステージングされたオブジェクトに関連する一定のオブジェクト情報290(UVなどの)はフィルタリングの決定のために保守される。このことは、図10を参照しながら以下に詳述するように、ステージングされたオブジェクトをページするかどうかについての後の決定に影響する。

【0049】図3に本発明の機能を備えたプロキシ・サーバ論理の例を示す。図示のように、ステップ310でプロキシ・サーバは入力待ち状態となる。ステップ315で、受信した入力によって、処置が分岐する。受信した入力(次の上位レベルから)がブッシュの場合、ステップ320でブッシュ・ハンドラ320が呼び出される。ブッシュ・ハンドラの詳細例は図5を参照しながら後述する。ステップ330で、受信した入力が欠落ブッシュ・オブジェクト要求の場合、欠落ブッシュ・オブジェクト要求ハンドラ240がステップ335で呼び出される。欠落ブッシュ・オブジェクト要求ハンドラ240は上位レベル・プロキシ・ノードによってフィルタリングされたオブジェクトに関して下位レベル・プロキシ(またはクライアント)からの要求を処理する。欠落ブッシュ・オブジェクト要求ハンドラの詳細例は図10を参照しながら後述する。ステップ350で、受信した入力がヘッド要求の場合(下位レベル・プロキシまたはクライアントからの)、ステップ360でヘッド要求ハンドラ260が呼び出される。ヘッド要求ハンドラ260はHTTPヘッド要求を処理して下位レベル・プロキシからのユーザ情報を送信する。ヘッド要求ハンドラの詳細例は図4を参照しながら後述する。ステップ350で、本発明の対象ではない他のタイプの入力について(従来のHTTP「フル」要求またはFTP要求など)、該当するミセニアス・ハンドラ(370)が呼び出される。

【0050】図4にヘッド要求ハンドラ(260)の例を示す。図示のように、ステップ410でプロキシは次の下位レベル・ノードiから受信したヘッド要求がヘッド内にユーザ・ラベル(ユーラベル)を含むかどうかをチェックする。ユーラベルは接続するカテゴリi値がそれぞれRVとPVで表される利用状況およびブリファレンスの2つのカテゴリを含む。次の下位レベル・ノードiのそれについて、プロキシ・サーバはメモリ内にその利用状況とブリファレンス・カテゴリ値をユーザ情報(280)のRV(i)とPV(i)内にそれぞれ保守

15

する。ステップ4 2 0で、 $RV(i)$ および $PV(i)$ はノード*i*の新たに受信した値に合わせて更新される。ステップ4 3 0で、プロキシ・ノードは総合利用状況とプリファレンス値(それぞれ $RVnode$ および $PVnode$ で示されている)をすべての次の下位レベルのノードにわたって保守する。これらの2つの平均方法を更新する際は指數平均法を用いることが好ましい。より詳細に言えば、 $PVnode$ はその現在の値プラス新しい $PV(i)$ と旧 $PV(i)$ 値の差の小数部分に設定される。この一例は小数部分が0.5である。 $RVnode$ 値も同様に設定される。ステップ4 4 0で、HTTPヘッダ中の他の情報を処理することができる。この一例はオブジェクトが最後に変更された時にに基づくオブジェクトの「新鮮度」をチェックすることである。

【0051】図5にブッシュ・ハンドラ2 2 0の例を示す。図示のように、ステップ6 1 0で次の(上位)レベルからブッシュされたオブジェクトが全オブジェクトの場合、すなわち、ヘッダだけではなく全コンテンツの場合、ステップ5 1 5でブッシュ・オブジェクト・フィルタリング・ルーチンが呼び出され、次の下位レベル・ノード(1つまたは複数)へのブッシュのフィルタリング決定がなされる。ブッシュ・オブジェクト・フィルタリング・ルーチンの詳細例は図6を参照しながら後述する。ステップ5 2 0で、このノードのオブジェクトをステージングするかどうかが決定するため、ステージング決定ルーチンが呼び出される。ステージング決定ルーチンの詳しい例については、図8に実験して述べる。ステップ5 1 0で、上位レベル・プロキシからサマリ情報だけがブッシュされた場合、ステップ5 2 5でブッシュ・サマリ・フィルタリング・ルーチンが呼び出され、サマリを下位レベル・ノードへブッシュするかどうかが決定される。ブッシュ・サマリ・フィルタリング・ルーチンの詳細例は図7を参照しながら後述する。ステップ5 3 0でブッシュ実行ルーチンが呼び出されて下位レベル・ノードへのブッシュが実行される。ブッシュ実行ルーチンの詳細例は図11を参照しながら後述する。

【0052】図6にブッシュ・オブジェクト・フィルタリング・ルーチンの例を示す。ステップ6 1 0で、インデックス変数*i*が0に初期化される。ステップ6 1 0で、インデックス変数*i*が次の下位レベルのノードの数Nより小さい場合、ステップ6 1 5で*i*の値が例えば1インクリメントし、ブッシュ決定変数 $P(i)$ が例えば0に初期化される。ステップ6 2 0で、オブジェクトのブッシュ(UV)の緊急度レベルが緊急ブッシュしきい値(PTII)より大きく、 $PV(i)$ が0より大きい場合(複数の下位レベルユーザーがオブジェクト内にプロファイルなどによってインタレストを指定した場合)、ステップ6 6 0でブッシュ決定が選択され($P(i)$ が1に設定され)、全オブジェクトがノード*i*へブッシュされる(図11のステップ1 2 0を参照)。そうでない

16

場合、ステップ6 3 0で、(論理) 関数 $F(RV(i), PV(i), UV(i), UV)$ 、すなわち、オブジェクトのプロパティ(例: UV)と、オブジェクトユーティリティ(例: $RV(i)$ および $PV(i)$)の関数が呼び出され、フィルタリング決定を行う。 $F(RV(i), PV(i), UV)$ が真の場合、ステップ6 6 0が実行され、 $P(i)$ は1に設定され、全オブジェクトがノード*i*へブッシュされる。 $F(RV(i), PV(i), UV)$ の簡単な例は論理式($(RV(i) UV > QTH) and (PV(i) > 0)$)である。ただし、QTHは1などのしきい値である。より複雑なF関数を設計してオブジェクトの帯域幅またはサイズあるいは両方の因数を考慮できる。例えば、別のF($RV(i), PV(i), UV)$ を $((RV(i) UV > WTH) and (PV(i) > 0))$ の論理式で表すことができる。ただし、WTHはオブジェクト・サイズと共に増加し、利用可能な帯域幅と共に減少するしきい値である。WTHはドキュメントの満了時間に考慮することもできる。この値を満了時間が長いオブジェクトに関して低く設定できる。ステップ6 4 0で、(論理) 関数 $G(RV(i), PV(i))$ が呼び出され、ブッシュ・サマリ決定が実行される。 $G(RV(i), PV(i))$ の例は論理式($(RV(i) > 0) and (PV(i) > 0)$)である。F関数と同様に、より複雑なG関数を設計して帯域幅などの他の因数をブッシュ・サーバの決定で考慮することができる。ステップ6 5 0で、 $P(i)$ は0.5に設定され、サマリは下方にノード*i*へブッシュされる。【0053】図7にブッシュ・サマリ・フィルタリング・ルーチンの例を示す。ステップ7 1 0で、インデックス変数*i*は0に初期化される。ステップ7 2 0で、インデックス変数*i*が次の下位レベルのノードの数Nより小さい場合、ステップ7 3 0で*i*の値が例えば1インクリメントする。ステップ7 4 0で、関数 $G(RV(i), PV(i))$ が呼び出され、ブッシュ・サマリ決定が実行される(これはステップ6 4 0に呼び出される関数と同じ因数である)。ステップ7 4 0で、G関数が真の場合、ステップ7 5 0で $P(i)$ は0.5に設定され、サマリは下方にノード*i*へブッシュされる(図11のステップ1 1 4 0を参照)。

【0054】図8にステージング決定ルーチンの例を示す。ステップ8 1 0で、オブジェクトOの「ステージング緊急度」因数が計算される。この因数はCacheU(O)として表される。オブジェクトのステージング緊急度の計算論理の例は図9を参照しながら以下に詳述する。ステップ8 1 5で、CacheU(O)の値は上位レベル・プロキシにあるオブジェクトOのステージング・データベースに基づいて下方に調整できる。ステージング・ステータス情報はHTTPヘッダのステージング・レベル(C-ラベル)に指定される。オブジェクトOがどこかの上位レベル・プロキシ・キャッシュですでにス

ステージングされている場合、それを現在のノードでステージングする必要は減少する。ステップ8 2 0で、`C a c h e U (O)` がより大きい場合、ステップ8 3 0で、空間量 (S) が (1) オブジェクトOよりもステージング緊急度が低いすべてのオブジェクトOjによって占められているか、(2) 利用可能、すなわち、現在使用されていないかが判定される。ステップ8 4 0でSがオブジェクトOのサイズ (O) より大きいと判定された場合、ステップ8 5 0でオブジェクトOのCV値が更新されて現在のノードでステージングされたことを示す。

詳細に言えは、元のCV値は2倍して1を加えて新しいCV値とすることができる。ステップ8 7 0で、オブジェクトOはキャッシュ (2 7 0) に記憶され、オブジェクトに関するCV、UV、および`C a c h e U`の値はメモリのオブジェクト情報 (2 9 0) 部分に記憶される。オブジェクトOは必要に応じて、ステージング緊急度がより低い値を持つ他のオブジェクトと交換できる。ステップ8 2 0で`C a c h e U (O)`の値がゼロの場合、オブジェクトはステージングされず、CV値は更新されてこのプロキシでステージングされていないことを示す。具体的には、元のCV値に2を掛けで新しいCV値を得ることができる。さらに、オブジェクトのC-ラベルはステップ8 5 0またはステップ8 6 0で生成された新しいCV値をとり、オブジェクトが隣居下位ノードにブッシュされる際のステージング・ステータスを示す (図11のステップ11 0 0を参照)。

【0 0 5 5】本発明の特徴と範囲を逸脱することなく、さまざまなものキャッシュ・組み込み処理が可能であることを当業者は理解するであろう。例えば、今まで要求がなかったとしても新しいカテゴリ内のオブジェクトを空間変数単位でステージングすることができる。

【0 0 5 6】図9にオブジェクトOのステージング緊急度の計算論理の例を示す。ステップ9 0 5で、ステージング緊急度の値`C a c h e U (O)`が`i`に初期化される。ステップ9 1 0で、インデックス変数iが0に初期化される。ステップ9 2 0で、インデックス変数iが次の下位レベルのノードNより小さく、ステップ9 3 0でブッシュ決定変数P (i) が1でない場合、ステップ9 4 0で`C a c h e U (O)`は`RV (i) UV (O)`だけインクリメントする。ステップ9 4 5で、iはインクリメントし、処理はステップ9 2 0に戻る。ステップ9 2 0でi>Nの場合、処理は終了する。

【0 0 5 7】図10に欠落ブッシュ・オブジェクト要求ハンドラ (2 4 0) の例を示す。ブッシュ・オブジェクトOがフィルタリングされ、その後下位レベル・ノードから要求された場合、ステップ1 0 0 5でオブジェクトOが現在のノードでステージングされているか判定される。ステージングされている場合、ステップ1 0 1 0でオブジェクトはそのC-ラベルにCV値を挿入して要求されたノードへ返送される。ステップ1 0 2 0で、オブ

ジェクトOのステージング緊急度の値が再び計算される。ステップ1 0 3 0で、この値がゼロまで低下した (オブジェクト上のインタレストを持つすべての下位レベルのノードがそのオブジェクトのコピーを受信した) 場合、または何か別の規定値もしくは計算値にまで低下した場合、ステップ1 0 4 0でオブジェクトOはもはやステージングされない。ステップ1 0 0 5で、オブジェクトがステージングされない場合、ステップ1 0 8 0で要求は上位レベル・プロキシ・サーバまたはコンテンツ送信元へ転送される。

【0 0 5 8】図11にブッシュ実行ルーチンの例を示す (図5、ステップ5 3 0)。ステップ1 1 0 0で、オブジェクトのCV値 (図8のステップ8 5 0または8 8 0から) がHTTPヘッダのC-ラベルに挿入される。ステップ1 1 2 0で、オブジェクトO全体がP (i) が1に等しいすべての次の下位レベルのノードへブッシュされる。ステップ1 1 4 0で、オブジェクトOのサマリ・ヘッダがP (i) が0、5のすぐれたレベルのすべてのノードにブッシュされる。ステップ1 1 6 0で、あるノードiへのブッシュが不成功の場合 (リンクまたはノード障害もしくはモービル・クライアントがアクセス不能)、P (i) 値がゼロにリセットされる。さらに、オブジェクトOのステージング決定ルーチン (図5のステップ5 2 0) がチェックされる。ステージング決定ルーチンがオブジェクトのステージングを実行しない場合、ステージング決定ルーチン (図8) は新しいP (i) 値のセットを指定して再呼び出され、ブッシュの一部が失敗したためにオブジェクトを今ステージングするかどうかを決定する。

【0 0 5 9】プロキシの一部がフィルタリング・プロトコルに適合せずコラボレーションに参加しない従来のプロキシである異機種プロキシ・サーバ環境では動的ブッシュ・フィルタリングが有効であることを当業者は理解するであろう。

【0 0 6 0】本発明の好ましい実施形態で、ウェブ・サーバの一般的なブッシュ・フィルタリング方式について説明してきた。ただし、本発明はブッシュ対象のオブジェクトが同様の特性を備えるいかなる種類の状況にも適用可能で、必ずしもインターネットやWWVNの適用分野に限定されないことを当業者は理解するであろう。

【0 0 6 1】本発明の好ましい実施形態は隣居の親ノードおよび子ノード間のステージングを伴う協動的なブッシュ・フィルタリングであるが、兄弟ノード間のコラボレーションを含めるように容易に適合させることができる。例えば、上位階層でプロキシが要求されたオブジェクトをステージングしていない場合、プロキシから兄弟プロキシへ権限会ができる。さらに、図8のステージング決定を、兄弟ノードを含め、だしこれに限定されず、階層外のプロキシ・ノードのステージング決定の因数を容易に適応させてそのステージング決定に使用すること

ができる。

【0062】まとめとして、本発明の構成に関して以下の事項を述べる。

【0063】(1) プッシュされたオブジェクトが階層中を下方に送信されるプロキシ階層においてその後プッシュされたオブジェクトを動的にフィルタリングする方法であって、プッシュされたオブジェクトに関する利用状況情報を階層中を上方に送信するステップと、送信されたオブジェクト利用状況情報を基づいてその後プッシュされたオブジェクトをフィルタリングするステップとを含む方法。

(2) プッシュされたオブジェクトが階層中を下方に送信されるプロキシ階層においてプッシュされたオブジェクトを動的にフィルタリングする方法であって、情報を総合しノード間で交換するステップと、プッシュされたオブジェクトを、総合され交換された情報に基づいてフィルタリングするステップとを含む方法。

(3) 前記フィルタリング・ステップがフィルタリングされたオブジェクトに関するメタ情報を送信するステップをさらに含む、上記(2)に記載の方法。

(4) フィルタリング決定、選択された下位レベル・プロキシ・クライアントへのプッシュの成功裏の完了、利用状況情報、および階層内の他のプロキシ・ノードでのステージング決定のいずれかに基づいてオブジェクトの適応ステージングを行うステップをさらに含む、上記(1)に記載の方法。

(5) プッシュされたオブジェクトがメタ情報のコンテンツ階層を含み、その後プッシュされたオブジェクトをフィルタリングする前記ステップがプロキシ階層中を下方にメタ情報を送信するステップをさらに含む、上記(1)に記載の方法。

(6) メタ情報を送信する前記ステップが、オブジェクトの短い説明を階層中を下方に送信するステップと、フィルタリングされたオブジェクトをプロキシ階層内でステージングするステップとをさらに含む、上記(5)に記載の方法。

(7) オブジェクトをステージングする前記ステップが、ユーザ・プロファイルで示された潜在的なインタレストを有するすぐ下位のレベルのすべてのプロキシまたはクライアント・ノードがオブジェクトを受信したか、あるいはステージング緊急度因数が所定および計算しきい値を下回ったか、そのいずれかの後にステージングされたオブジェクトをページするステップをさらに含む、上記(4)に記載の方法。

(8) 前記フィルタリング・ステップがさらに階層内のすべての下位クライアント・ノードによるプッシュされたオブジェクトの総合利用状況およびブリファレンスを規定するステップをさらに含む、上記(1)に記載の方法。

(9) オブジェクトがオブジェクト・グループに分類さ

れ、各オブジェクト・グループの利用状況情報を以前のユーザ要求パターンに基づく、上記(4)に記載の方法。

(10) 利用状況情報が下位レベル・プロキシまたはクライアントの要求パターンを含み、前記プッシュされたオブジェクトを異なるグループに分類するステップと、プッシュされたオブジェクトのグループ分けを含む利用状況情報を総合しノード間で交換するステップと、総合され交換された情報およびプッシュされたオブジェクトのグループ化に基づいてプッシュされたオブジェクトをフィルタリングするステップとをさらに含む、上記(1)に記載の方法。

(11) フィルタリング決定またはステージング決定が審査幅、オブジェクト・プロパティ、またはクライアント特性のいずれかの関数である、上記(4)に記載の方法。

(12) クライアント特性がユーザ・プロファイルまたはブリファレンス情報を含む、上記(11)に記載の方法。

(13) オブジェクト緊急度標識をプッシュされたオブジェクトに関する階層中を下方に送信するステップをさらに含み、前記フィルタリング・ステップがオブジェクト緊急度標識の関数である、上記(1)に記載の方法。

(14) 緊急度標識を1つまたは複数の異なる階層レベルに関連付けるステップをさらに含み、前記ステージング・ステップが、緊急度標識の関数としてコンテンツ階層の1つまたは複数のレベルでオブジェクトをステージングするステップを含む、上記(4)に記載の方法。

(15) 前記フィルタリング・ステップとステージング・ステップのいずれかがオブジェクト・サイズの関数である、上記(4)に記載の方法。

(16) 前記フィルタリング・ステップと前記ステージング・ステップのいずれかがオブジェクトの寿命または満了時間の関数である、上記(4)に記載の方法。

(17) 前記ステージング・ステップに応答して、プッシュされたオブジェクトのステージング・ステータスをプッシュされたオブジェクトへ送信するステップをさらに含む、上記(4)に記載の方法。

(18) プロキシ階層が、前記フィルタリング・ステップと前記ステージング・ステップのいずれかが差異内のすべてのサーバによっては実行されない異種プロキシ階層を含む、上記(4)に記載の方法。

(19) PICS利用状況ラベルを作成し、PICSカテゴリー値を使ってプロキシ階層の下位レベルでの総合利用状況情報を表すステップをさらに含み、前記送信ステップがPICS利用状況ラベルを使って総合利用状況情報を階層中を上方に送信するステップを含む、上記(2)に記載の方法。

(20) PICSステージング・ラベルを作成し、P I

C S カテゴリ値を使ってプロキシ階層の所のレベルにおけるステージングされたオブジェクトのステージング
・ステータスを表すステップと、P I C S ステージング
・ラベルを使ってステージング・ステータスを階層中を下方に送信するステップをさらに含む、上記（4）に記載の方法。

（2 1）P I C S ブッシュ・ラベルを作成し、P I C S カテゴリ値を使ってブッシュされたオブジェクトの緊急度標識を表すステップと、P I C S ブッシュ・ラベルを使って緊急度標識を階層中を下方に送信するステップとをさらに含む、上記（4）に記載の方法。

（2 2）メタ情報プロトコルを使って階層中を情報を送信する、上記（1）ないし（3）、（5）、（6）、（9）ないし（10）、（14）ないし（15）、（19）ないし（20）または（21）のいずれか一項に記載の方法。

（2 3）P I S C プロトコルを使って階層中を情報を送信する、上記（1）ないし（3）、（5）、（6）、（9）ないし（10）、（14）ないし（15）、（19）ないし（20）または（21）のいずれか一項に記載の方法。

（2 4）P I C S ブッシュ・ラベルを作成し、P I C S カテゴリ値を使ってブッシュされたオブジェクトのサマリを表すステップと、P I C S ブッシュ・ラベルを使ってサマリを階層中を下方に送信するステップとをさらに含む、上記（4）に記載の方法。

（2 5）階層外のプロキシ・ノードについてのステージング決定に基づいてオブジェクトの適応ステージングを行なうステップをさらに含む、上記（1）に記載の方法。

（2 6）ブッシュされたオブジェクトのストリームが階層中を下方に送信されるプロキシ階層において、その後ブッシュされるオブジェクトを動的にフィルタリングする方法であって、ブッシュ・ストリームにメタ情報の注釈を付けるステップと、前記の注釈付ステップに応答

して、1つまたは複数のブッシュされたオブジェクトをフィルタリングするステップとを含む方法。

（2 7）フィルタリング決定とステージング決定のいずれかが、コンテンツ階層のレベルが異なると変わる、上記（6）に記載の方法。

【図面の簡単な説明】

【図 1】本発明の機能を備えた上位レベル・クライアントサーバ・アーキテクチャの図である。

【図 2】図 1 のサーバの例を示す図である。

【図 3】サーバ論理の例を示す図である。

【図 4】ヘッド要求ハンドラの例を示す図である。

【図 5】ブッシュ・ハンドラの例を示す図である。

【図 6】ブッシュ・オブジェクト・フィルタリング・ルーチンの例を示す図である。

【図 7】ブッシュ・サマリ・フィルタリング・ルーチンの例を示す図である。

【図 8】ステージング決定ルーチンの例を示す図である。

【図 9】ステージング緊急度の計算例を示す図である。

【図 10】欠落したブッシュ・オブジェクト要求ハンドラの例を示す図である。

【図 11】ブッシュ実行ルーチンの例を示す図である。

【符号の説明】

2 0 ウェブ・サイト

2 5 インターネット

3 0 レベル 0 プロキシ

3 5 レベル 1 プロキシ

3 7 レベル 1 プロキシ

4 0 レベル 2 プロキシ

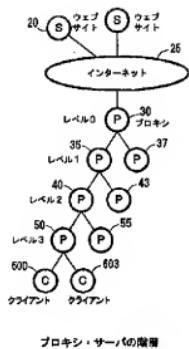
3 0 4 3 レベル 2 プロキシ

5 0 レベル 3 プロキシ

6 0 0 クライアント

6 0 3 クライアント

【図1】



【図2】

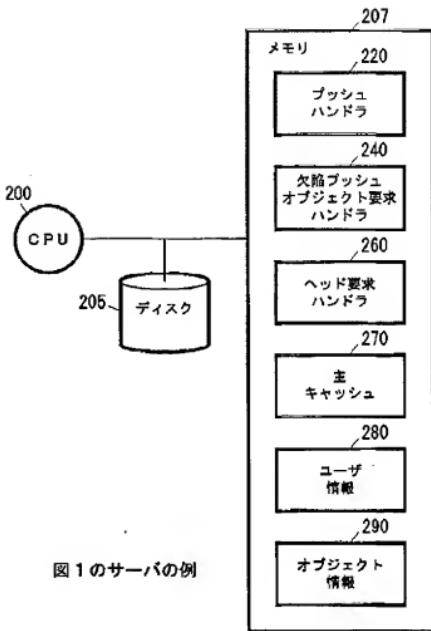
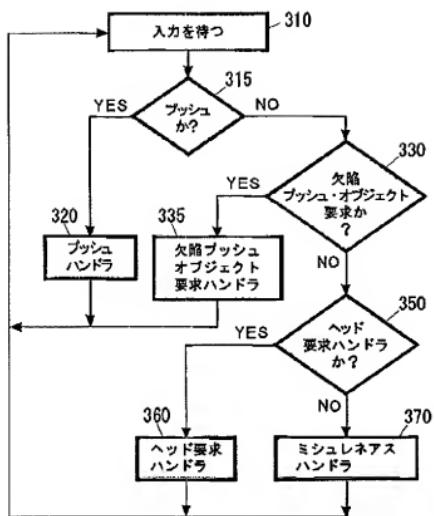
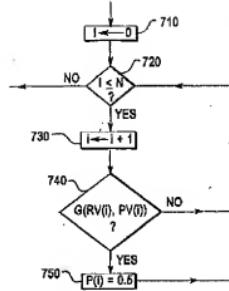


図1のサーバの例

【図3】



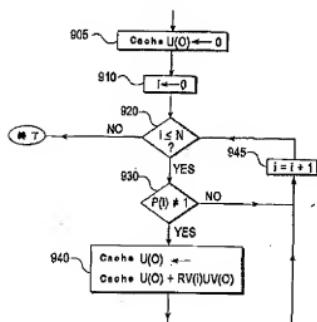
【図7】



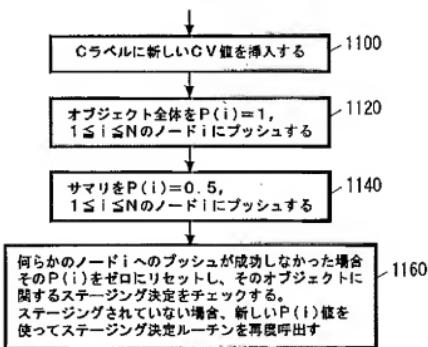
プッシュ・サマリ・フィルタリング・ルーチン

サーバ論理の例

【図9】



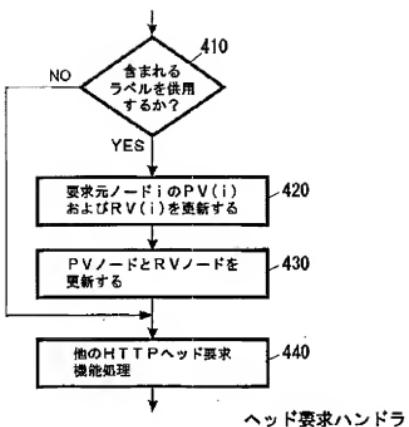
【図11】



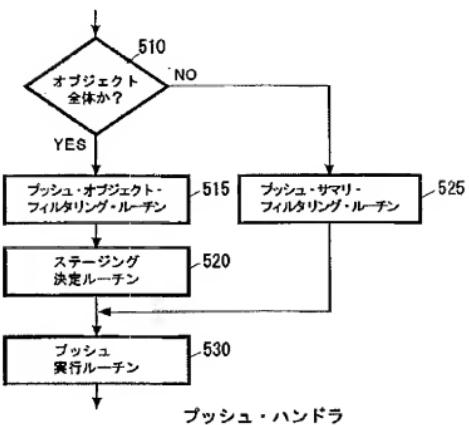
ステージング緊急度の計算

プッシュ実行ルーチン

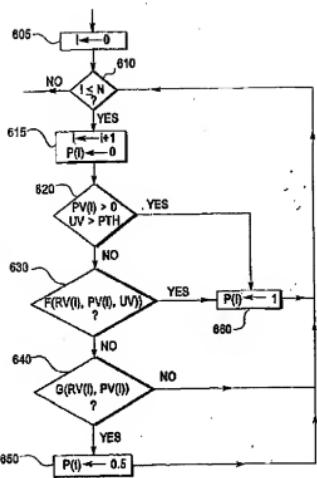
【図4】



【図5】

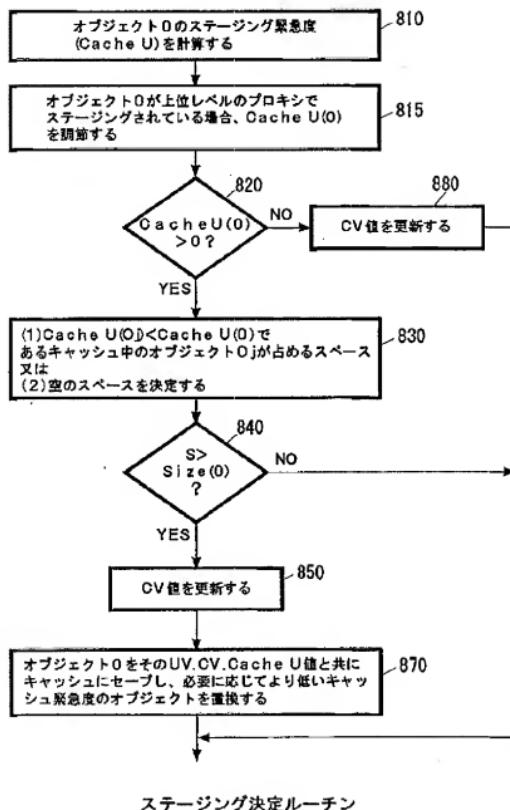


【図6】

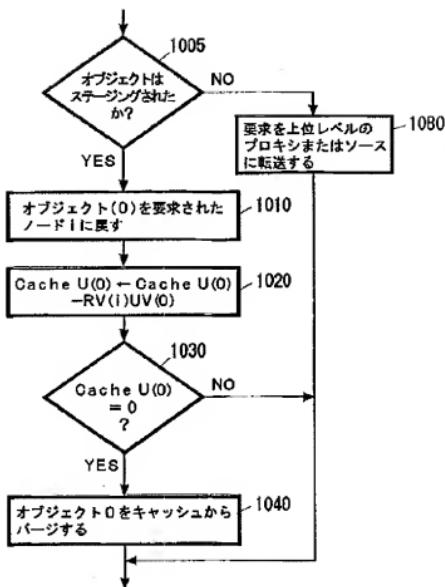


ブッシュ・オブジェクト・フィルタリング・ルーチン

【図8】



【図10】



欠陥プッシュ・オブジェクト要求ハンドラ

フロントページの続き

(72) 発明者 ピーター・ケニス・マルキン
 アメリカ合衆国10502 ニューヨーク州ア
 ーズレイ ブランブル・ブルック・ロード
 64

(72) 発明者 ロバート・ジェフリー・シユロス
 アメリカ合衆国10510 ニューヨーク州ブ
 ライアークリフ・マナー ホルブルック・
 レーン 155

(72) 発明者 フィリップ・シーニルン・ユー
 アメリカ合衆国10514 ニューヨーク州チ
 ャバクア ストーノウェイ 18